PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-293720
(43)Date of publication of application : 04.11.1998

(51)Int.Cl. G06F 12/08

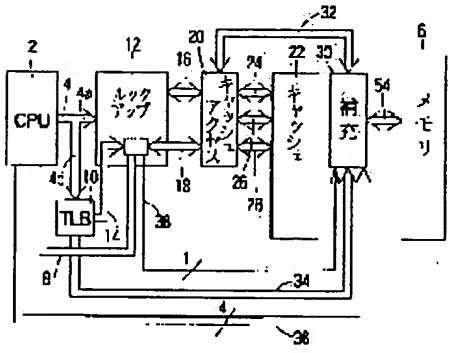
(21)Application number : 10-053904
(22)Date of filing : 05.03.1998

(71)Applicant : SGS THOMSON MICROELECTRON LTD
(72)Inventor : BARNABY CATHERINE
FARRALL GLENN
FEL BRUNO

(30)Priority
Priority number : 97 9704542 Priority date : 05.03.1997 Priority country : GB

(54) CACHE COHERENCY MECHANISM IN COMPUTER SYSTEM, AND METHOD FOR CHANGING COHERENCY STATE OF CACHE CONTENT ABOUT ITEM HELD IN MAIN MEMORY

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a cache coherency mechanism in a computer system and a method which changes the coherency state of cache content about an item that is held in main memory.
SOLUTION: A computer system has a processor, a cache and main memory. A cache coherency mechanism offers a cache coherency instruction which separately designates an operation that should be executed to the content of a storage place in the cache 22 and an address in the main memory 6. With this, the content of the cache 22 becomes coherent to the memory 6. When an executing process gives a normal access to the address in the memory 6, the operation is carried out to the content of the storage place in the cache when it is replenished by accessing the address in the memory regardless of whether the content of a designated address in the memory 6 is held in the storage place or not.



LEGAL STATUS

[Date of request for examination]	26.03.1998
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	2968509
[Date of registration]	20.08.1999
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平10－293720

(43)公開日　平成10年(1998)11月４日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　12/08	３１０	Ｇ０６Ｆ　12/08	３１０Ｂ

審査請求　有　　請求項の数13　ＯＬ　（全 13 頁）

(21)出願番号　特願平10－53904

(22)出願日　平成10年(1998)３月５日

(31)優先権主張番号　９７０４５４２．１

(32)優先日　1997年３月５日

(33)優先権主張国　イギリス（ＧＢ）

(71)出願人　595008364
エスジーエス・トムソン、マイクロエレクトロニクス、リミテッド
ＳＧＳ－ＴＨＯＭＳＯＮ　ＭＩＣＲＯＥＬＥＣＴＲＯＮＩＣＳ　ＬＴＤ.
イギリス国ブリストル、アーモンズベリー、アズテック、ウエスト、1000

(72)発明者　キャサリン、バーナビー
イギリス国ブリストル、コールピット、ヒース、ラウンドウェイズ、18

(74)代理人　弁理士　佐藤　一雄　（外３名）

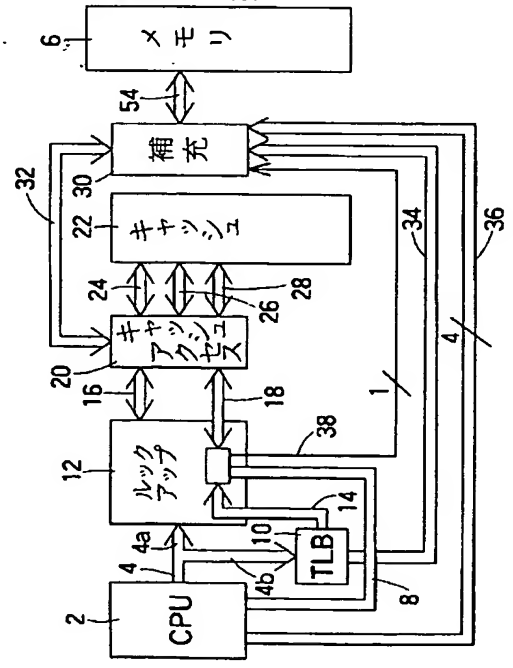
最終頁に続く

(54)【発明の名称】　コンピュータシステムにおけるキャッシュ・コヒーレンシー機構および主メモリに保持されている項目に関してキャッシュの内容のコヒーレンシー状態を変更する方法

(57)【要約】

【課題】　コンピュータシステムにおけるキャッシュ・コヒーレンシー機構および主メモリに保持されている項目に関してキャッシュの内容のコヒーレンシー状態を変更する方法を得ること。

【解決手段】　コンピュータシステムがプロセッサと、キャッシュと、主メモリとを有する。キャッシュ・コヒーレンシー機構によって、１）キャッシュ２２内の記憶場所の内容に対して実行すべきオペレーションと、２）主メモリ６内のアドレスと、をおのおの指定するキャッシュ・コヒーレンシー命令を提供することにより、キャッシュ２２の内容を主メモリ６に関してコヒーレントにする。実行しているプロセスが主メモリ６内のそのアドレスを通常アクセスすると、主メモリ６内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、主メモリ６内のそのアドレスをアクセスすることにより補充できたときキャッシュ内の記憶場所の内容に対してオペレーションが実行される。



【特許請求の範囲】

【請求項１】プロセッサと、キャッシュと、主メモリとを備え、主メモリ内の複数のアドレスがキャッシュ内の各記憶場所をアクセスし、プロセッサにより実行されるプロセスが、（ｉ）キャッシュ内の記憶場所の内容に対して実行すべきオペレーションと、（ｉｉ）主メモリ内のアドレスと、を指定するキャッシュ・コヒーレンシー命令を含み、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスしたとき、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、主メモリ内の前記アドレスに対するアクセスにより充たすことができるキャッシュ内の記憶場所の内容に対してオペレーションが実行される、コンピュータシステムにおけるキャッシュ・コヒーレンシー機構。

【請求項２】請求項１記載のキャッシュ・コヒーレンシー機構であって、各キャッシュ記憶場所の内容は主メモリ内のアドレスと、主メモリ内のそのアドレスに記憶されている項目とを含むキャッシュ・コヒーレンシー機構。

【請求項３】請求項２記載のキャッシュ・コヒーレンシー機構であって、キャッシュ・コヒーレンシー命令が、そのキャッシュ記憶場所に保持されている、主メモリ内のアドレスに、キャッシュ内の前記アドレスに保持されている項目をライトバックするフラッシュ命令であるキャッシュ・コヒーレンシー機構。

【請求項４】請求項１または２記載のキャッシュ・コヒーレンシー機構であって、キャッシュ・コヒーレンシー命令は、キャッシュ内の前記記憶場所の内容をクリアする除去命令であるキャッシュ・コヒーレンシー機構。

【請求項５】請求項１ないし４のいずれかに記載のキャッシュ・コヒーレンシー機構であって、キャッシュ・コヒーレンシー命令は主メモリ内の一連のアドレスを指定し、前記一連のアドレス中のアドレスをアクセスすることにより通常充たされる、キャッシュ内の１組の記憶場所の内容のために動作するキャッシュ・コヒーレンシー機構。

【請求項６】請求項１ないし５のいずれかに記載のキャッシュ・コヒーレンシー機構であって、キャッシュは複数のキャッシュ区画に区分され、キャッシュ内の関連する記憶場所を含むキャッシュ区画は主メモリ内の指定されたアドレスに依存して決定されるキャッシュ・コヒーレンシー機構。

【請求項７】請求項１ないし６のいずれかに記載のキャッシュ・コヒーレンシー機構であって、主メモリはページで編成され、各ページは一連のアドレスを含み、キャッシュ・コヒーレンシー命令は主メモリ内のページのうち、オペレーションを実行すべきページを指定し、オペレーションは指定されたページ内のその一連の各アドレスに対して実行されるキャッシュ・コヒーレンシー機構。

【請求項８】請求項６または７記載のキャッシュ・コヒーレンシー機構であって、各ページ内のアドレスの数は、前記キャッシュ区画の１つにおける記憶場所の数と常に少なくとも同じであるキャッシュ・コヒーレンシー機構。

【請求項９】請求項１ないし８のいずれかに記載のキャッシュ・コヒーレンシー機構であって、キャッシュ、または存在する時は各キャッシュ区画が直接マップされるキャッシュ・コヒーレンシー機構。

【請求項１０】一連の命令を実行することによりプロセスを実行するプロセッサと、

前記命令、および前記命令に対するデータを保持する主メモリと、

プロセッサと主メモリとの間のメモリアクセス経路中に接続され、複数の記憶場所を有するキャッシュとを備え、主メモリ内の複数のアドレスが各記憶記憶場所をアクセスし、

プロセッサが実行するための一連の命令は、（ｉ）キャッシュ内の記憶場所の内容に対して実行すべきオペレーションと、（ｉｉ）主メモリ内のアドレスと、を指定するキャッシュ・コヒーレンシー命令を含み、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスしたとき、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、指定されたオペレーションは、主メモリ内の前記指定されたアドレスに対するアクセスにより充たすことができるキャッシュ内の記憶記憶場所の内容に対して実行される、コンピュータシステム。

【請求項１１】請求項１０記載のコンピュータシステムであって、プロセッサはユーザー動作モードと、特権動作モードとを有し、キャッシュ・コヒーレンシー命令はユーザーモードで実行される、コンピュータシステム。

【請求項１２】主メモリ内の複数のアドレスがキャッシュの各記憶場所をアクセスする、主メモリに保持されている項目に関してキャッシュの内容のコヒーレンシー状態を変更する方法であって、

（ｉ）キャッシュ内の記憶場所の内容を実行すべきオペレーションと、（ｉｉ）主メモリ内のアドレスとを指定するキャッシュ・コヒーレンシー命令を実行する過程と、

前記キャッシュ・コヒーレンシー命令に応じて、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスしたとき、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、主メモリ内の前記アドレスに対するアクセスにより充たすことができるキャッシュ内の記憶場所の内容に対する指定されたオペレーションを実行する過程とを備える、主メモリに保持されている項目に関してキャッシュの内容のコヒーレンシー状態を変更する方

法.

【請求項13】（ｉ）キャッシュ内の記憶場所の内容について実行すべきオペレーションと、（ii）主メモリ内のアドレスと、を指定するキャッシュ・コヒーレンシー命令を含み、キャッシュ・コヒーレンシー命令は、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスする場合にのみ、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、主メモリ内の前記アドレスに対するアクセスにより充たすことができるキャッシュ内の記憶場所の内容に対して、指定されたオペレーションを実行させる命令セット。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はキャッシュ・コヒーレンシー機構に関するものである。

【０００２】

【従来の技術】この技術で周知のように、あるデータおよびコードに対するアクセス待ち時間を短縮し、かつそのデータおよびコードのために使用するメモリの帯域幅を狭くするために、コンピュータシステムにおいてキャッシュメモリが用いられる。キャッシュメモリはメモリアクセスを遅らせ、まとめ、かつ集合させることができる。

【０００３】キャッシュメモリはコンピュータのプロセッサと主メモリとの間で動作する。プロセッサで実行しているプロセッサにより求められるデータと命令のデータとの少なくとも一方を、そのプロセスの実行中にキャッシュに保持することができる。キャッシュのアクセスは主メモリのアクセスより通常はるかに迅速である。プロセッサがキャッシュメモリ内の求められているデータ項目または命令項目を見つけないときは、プロセッサは主メモリを直接にアクセスしてそれを検索し、求められているデータ項目または命令項目がキャッシュにロードされる。キャッシュメモリを使用および再び充たすための種々の既知のシステムが存在する。

【０００４】実時間システムでキャッシュに依存するためには、キャッシュの動作を予測できる必要がある。すなわち、キャッシュ内で見出されることが予測される特定のデータ項目または特定の命令が、実際にそこに見出される合理的な確実性に富む必要がある。既存の再充填機構のほとんどは、求められているデータ項目または命令をキャッシュ内に置くことを通常つねに試みる。これを行うために、既存の機構は他のデータ項目または他の命令をキャッシュから削除しなければならない。そうすると、後で使用するために存在することが予測されている項目が削除される結果となることがある。これは、多重タスキングの場合、または割り込みプロセスあるいはその他の予測できないプロセスを取り扱わなければならないプロセッサの場合に、とくにそうである。

【０００５】コンピュータシステムは２つ以上のプロセッサを持つことがあり、各プロセッサはそれ自身のキャッシュを持つことができる。あるいは、プロセッサは複数のＣＰＵを持つことができる。各ＣＰＵはそれ自身のキャッシュを有する。しかし、それらのキャッシュは単一のメモリ資源を普通にアクセスする。

【０００６】図7は、それ自身のキャッシュ１とキャッシュ２をおのおの持つ２つのプロセッサＣＰＵ１とＣＰＵ２がある場合を示す。それらのキャッシュは１つのメモリ資源ＭＥＭを共有する。図8はそのような状況において起きることがあるものを示す。主メモリ内のアドレス１０１０について考えることにする。これはキャッシュ１と２のキャッシュ記憶場所１０にマップされる。アドレス１０１０に記憶されている値Ｖ３は初期値Ｘを持っていた。値Ｖ３＝Ｘは最初は両方のキャッシュのキャッシュ記憶場所に記憶されていた。その段階では、データ項目Ｖ３は「見ることが」できた、すなわち、アドレス１０１０をアクセスしているいずれのプロセッサもそれのキャッシュから値Ｖ３＝Ｘを検索する。しかし、ＣＰＵ１はプロセスを実行し、値Ｖ３＝Ｘを変更し、これをＣＡＳＨ１の記憶場所１０へ戻していた。今は主メモリ内の値Ｖ３＝Ｘは「汚れている」－すなわち、それはＶ３の現在の値をもはや反映しない。更に、キャッシュ２内の値Ｖ３＝Ｘは「古い」、すなわち、それは真の値とは異なる。この状況は、ＣＰＵ２が値Ｖ３を検索しようとする前に訂正する必要がある。その理由は、もし訂正しないとＶ３＝Ｘを誤って検索するからである。

【０００７】このようにして、いくつかのプロセッサおよび装置がメモリを正しく共用できるようにするために、キャッシュ・コヒーレンシー制御が求められる。これは下記のようにして行うことができる。

1．自動コヒーレンシー。ハードウェアを追加することにより、どのプロセッサまたはどの装置が書き込んだかとは無関係に、最後に書き込まれた値をロード（loads）が検索できることが保証される。自動コヒーレンシーの機能的ではあるが、性能が低い、実現はキャッシュを不能にすることであることに注目されたい。そのような追加のハードウェアを図7に参照記号 COHERE で示す。

2．ソフトウエア・コヒーレンシー。キャッシュとメモリとの間のデータの転送を制御するために特殊なコード列を用いる。それらのコード列によりコヒーレンシーの精密な制御と、キャッシュの効率的な使用が可能になる。

【０００８】データの可視性はキャッシュが自動的にコヒーレントであるか否かに依存する。キャッシュが自動的にコヒーレントでないとすると、メモリの内容と、そのメモリの自身のキャッシュの内容だけがプロセッサにとって見える。適切な時にデータがメモリに確実に書込まれるようにするために、ソフトウエアは協力しなけれ

ばならない。キャッシュが自動的にコヒーレントであれば、任意のプロセッサにより最後に書込まれた値を他の全てのプロセッサが見ることができる。

【0009】<可視性の定義>

可視　データ項目のアドレスからのロードがその項目を戻すならば、そのデータ項目をプロセッサが見ることができる。

古い　キャッシュ内の値が最後に書込まれた値とは異なるならば、データ項目は古い。

汚れ　データ項目が主メモリに関してキャッシュ内で修正されたならば、そのデータ項目は汚れている。

【0010】キャッシュ内の記憶場所をクリアすることをプロセスが希望するが、そのプロセスはそのキャッシュ記憶場所に記憶されているアドレスをアクセスせず、既存のソフトウエア・コヒーレンシー技術が、カーネルモードと呼ばれる特殊な、特権プロセッサ動作モードの使用を求める。正常なユーザーモードでは、カーネルモードへの転送によること以外のソフトウエア・コヒーレンシー技術を用いて、キャッシュをコヒーレントにすることがそのような状況においては可能でない。

【0011】

【発明の概要】本発明の1つの面によれば、プロセッサと、キャッシュと、主メモリとを備え、主メモリ内の複数のアドレスがキャッシュ内の各記憶場所をアクセスし、プロセッサにより実行されるプロセスが、（ｉ）キャッシュ内の記憶場所の内容に対して実行すべきオペレーションと、（ii）主メモリ内のアドレスと、を指定するキャッシュ・コヒーレンシー命令を含み、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスしたとき、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、オペレーションは主メモリ内の前記アドレスに対するアクセスにより充たすことができるキャッシュ内の記憶場所の内容に対して実行される、コンピュータシステムにおけるキャッシュ・コヒーレンシー機構が得られる。

【0012】各キャッシュ記憶場所の内容はメモリ内のアドレスと、主メモリ内のそのアドレスに記憶されている項目とを含むことができる。主メモリ内のアドレスの全てまたは一部を保持することができる。項目はデータ項目または命令とすることができる。

【0013】上記キャッシュ・コヒーレンシー機構は、キャッシュに記憶されている特定のアドレスについてキャッシュ・コヒーレンシーオペレーションを実行することを要求する必要がないという利点を持つ。そのキャッシュ記憶場所にマップする任意のアドレスを指定でき、プロセッサは命令がそのアドレスを通常アクセスするならばその命令を実行することができる。したがって、実行しているプロセスがキャッシュ・コヒーレンシー命令の主メモリ内の指定されたアドレスをアクセスしなければキャッシュ・コヒーレンシー・オペレーションが実行されないから、どのような保護モードも自動的に考慮に入れる。

【0014】1つの種類のキャッシュ・コヒーレンシー命令は、そのキャッシュ記憶場所に保持されている主メモリ内のアドレスに、キャッシュ記憶場所に保持されている項目をライトバックするフラッシュ命令である。

【0015】他の種類のキャッシュ・コヒーレンシー命令は、そのキャッシュ記憶場所の内容をクリアする排除命令である。

【0016】キャッシュ・コヒーレンシー命令は主メモリ内の一連のアドレスを指定でき、かつその一連のアドレス中のアドレスをアクセスすることにより通常充たされるキャッシュ内の1組の記憶場所の内容について動作する。あるいは、主メモリ内の1つのアドレスをおのおの指定する一連のキャッシュ・コヒーレンシー命令を実行することができる。

【0017】キャッシュを複数のキャッシュ区画に区分することができる。キャッシュ内の関連する記憶場所を含むキャッシュ区画が、主メモリ内の指定されたアドレスに依存して決定される。特定のキャッシュ区画の実現のこれ以上の詳細は我々の先願明細書に記載されている。

【0018】主メモリはページで編成することができる。各ページは一連のアドレスを含む。その場合には、キャッシュ・コヒーレンシー命令はオペレーションを実行すべき主メモリ内のページを指定することができる。オペレーションは指定されたページ内の一連のアドレスのおのおのについて実行される。

【0019】その場合には、各ページ内のアドレスの数がキャッシュ区画の1つにおける記憶場所の数より常に大きいならば、ページを指定することによりキャッシュ区画を常に完全にクリヤできることを決定することができる。

【0020】キャッシュ、またはキャッシュ区画は直接マップすることができる。しかし、他の結合（associativities）が可能である。

【0021】本発明は、一連の命令を実行することによりプロセスを実行するプロセッサと、前記命令と、前記命令に対するデータとを保持する主メモリと、プロセッサと主メモリとの間のメモリアクセス経路中に接続され、複数の記憶場所を有するキャッシュと、を備え、主メモリ内の複数のアドレスが各記憶場所をアクセスし、プロセッサが実行するための一連の命令は、（ｉ）キャッシュ内の記憶場所の内容に対して実行すべきオペレーションと、（ii）主メモリ内のアドレスと、を指定するキャッシュ・コヒーレンシー命令とを含み、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスしたとき、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは

無関係に、指定されたオペレーションは、主メモリ内の前記指定されたアドレスに対するアクセスにより充たすことができるキャッシュ内の記憶場所の内容に対して実行される、コンピュータシステムも提供する。

【００２２】本発明は、主メモリ内の複数のアドレスがキャッシュの各記憶場所をアクセスする、主メモリに保持されている項目に関してキャッシュの内容のコヒーレンシー状態を変更する方法であって、（ｉ）キャッシュ内の記憶場所の内容を実行すべきオペレーションと、（ｉｉ）主メモリ内のアドレスとを指定するキャッシュ・コヒーレンシー命令を実行する過程と、前記キャッシュ・コヒーレンシー命令に応じて、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスしたとき、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、主メモリ内の前記アドレスに対するアクセスにより充たすことができるキャッシュ内の記憶場所の内容に対する指定されたオペレーションを実行する過程とを備える、主メモリに保持されている項目に関してキャッシュの内容のコヒーレンシー状態を変更する方法も提供する。

【００２３】本発明は、（ｉ）キャッシュ内の記憶場所の内容について実行すべきオペレーションと、（ii）主メモリ内のアドレスと、を指定するキャッシュ・コヒーレンシー命令を含み、キャッシュ・コヒーレンシー命令は、実行しているプロセスが主メモリ内の前記アドレスを通常アクセスする場合にのみ、主メモリ内の指定されたアドレスの内容がキャッシュ内のその記憶場所に保持されているか否かとは無関係に、主メモリ内の前記アドレスに対するアクセスにより充たすことができるキャッシュ内の記憶場所の内容に対して、指定されたオペレーションを実行させる命令セットも更に提供する。

【００２４】好適な実施の形態では、プロセッサはユーザー動作モードと、特権（カーネル）動作モードとを有する。キャッシュ・コヒーレンシー命令はユーザーモードで実行することができる。

【００２５】

【発明の実施の形態】以下、図面を参照して本発明を実施の形態について説明する。

【００２６】図１はキャッシュ装置を組み込んだコンピュータのブロック図である。このコンピュータはＣＰＵ２を有する。そのＣＰＵ２は、主メモリ６から項目をアクセスするためにアドレスバス４に接続され、かつ項目をＣＰＵ２に戻すためにデータバス８に接続される。データバス８をここではデータバスと呼んでいるが、それは、主メモリ６からの項目が実際のデータ、またはＣＰＵが実行する命令を構成しようが、構成しまいが、それらの項目を戻すためのものであることがわかるであろう。ここで説明する装置は命令キャッシュおよびデータキャッシュに使用するために適当なものである。よく知られているように、データキャッシュと命令キャッシュを別々に設けることもできれば、データキャッシュと命令キャッシュを組み合わせることもできる。ここで説明しているコンピュータでは、アドレッシングのやり方はいわゆる垂直アドレッシング法である。アドレスはラインインページ・アドレス４ａと、垂直ページアドレス４ｂとに分割される。垂直ページアドレス４ｂは翻訳ルックアサイド・バッファ（ＴＬＢ）１０に供給される。ラインインページ・アドレス４ａはルックアップ回路１２に供給される。翻訳ルックアサイド・バッファ１０は、垂直アドレス４ｂから変換された実ページアドレス１４をルックアップ回路１２に供給する。ルックアップ回路１２はアドレスバス１６およびデータバス１８を介してキャッシュアクセス回路２０に接続される。また、データバス１８は主メモリ６からのデータ項目または命令のためのものにすることができる。キャッシュアクセス回路２０はアドレスバス２４と、データバス２６と、制御バス２８とを介してキャッシュメモリ２２に接続される。制御バス２８はキャッシュメモリのための交換情報を転送する。補充装置（refill engine）３０が補充バス（refill bus）３２を介してキャッシュアクセス回路２０に接続される。補充バス３２は交換情報と、データ項目（または命令）と、アドレスとを補充装置とキャッシュアクセス回路との間で転送する。補充装置３２自体は主メモリ６に接続される。

【００２７】補充装置３０はフル実アドレス３４を翻訳ルックアサイド・バッファ１０から受ける。フル実アドレス３４は主メモリ６内の項目の実ページアドレスと、ラインインページ・アドレスとを含む。また、補充装置３０は翻訳ルックアサイド・バッファ１０からの区画標識を４ビットバス３６を介して受ける。区画標識の機能については後で説明する。

【００２８】最後に、補充装置３０はミス信号を線３８を介して受ける。ミス信号は、後で詳しく説明するやり方でルックアップ回路１２で発生される。

【００２９】ここで説明しているキャッシュメモリ２２は直接マップされるキャッシュである。すなわち、それは複数のアドレス可能な記憶場所を有する。各記憶場所はキャッシュの１つの行を構成する。各行は主メモリからの項目と、その項目の主メモリ内のアドレスの部分とを含む。各行に記憶されているデータ項目の主メモリ内のアドレスの最下位ビットを表すビットの数により構成された行アドレスにより、各行はアドレスすることができる。たとえば、８つの行があるキャッシュ・メモリでは、それらの行を一意に特定するために各行アドレスは３ビット長である。たとえば、キャッシュの第２の行の行アドレスは００１であるから、ビット００１で終わるアドレスを持つ主メモリからの任意のデータを保持することができる。明らかに、主メモリでは、そのようなアドレスが多数存在するために、キャッシュメモリ内のそ

の行に保持されるデータ項目が潜在的に多数存在する。もちろん、キャッシュメモリはその行には1度にただ1つのデータ項目を保持することができる。

【0030】ここで図1に示すコンピュータシステムの動作を説明するが、説明に際しては区画標識が存在しないものとする。CPU2は主メモリ6内のアドレスを用いて主メモリから項目を要求し、そのアドレスをアドレスバス4を介して送る。仮想ページ数が翻訳ルックアサイド・バッファ10に供給される。そのバッファはそのページ数を、所定の仮想－実ページ翻訳アルゴリズムに従って実ページ数14に翻訳する。実ページ数14は、CPU2により送られた元のアドレスのラインインページ数4aとともにルックアップ回路12に供給される。キャッシュアドレス回路20はキャッシュメモリ22をアドレスするためにラインインページ・アドレスを用いる。ラインインページ・アドレスは、キャッシュメモリ22内の行アドレスに等しいメモリ内の主メモリの1組の最下位ビット（終りのビットを必ずしも含まない）を含む。ラインインページ・アドレスにより特定された行アドレスにおけるキャッシュメモリ22の内容（データ項目（または命令）とデータ項目（または命令）の主メモリ内のアドレスである）がルックアップ回路12に供給される。そこで、キャッシュメモリから検索されたアドレスの実ページ数を、翻訳ルックアサイド・バッファ10から供給された実ページ数と比較する。それらのアドレスが一致したならば、キャッシュメモリのその行に保持されていたデータ項目をデータバス8に沿ってCPUへ戻させるヒットをルックアップ回路は指示する。しかし、キャッシュメモリ22内のアドレスされた行に保持されていたアドレスの実ページ数が、翻訳ルックアサイド・バッファ10から供給された実ページ数に一致しないとすると、補充装置30への線38にミス信号が発生される。翻訳ルックアサイド・バッファ10からバス34を通じて供給される実アドレスを用いて、主メモリ6から正しい項目を検索することが補充装置30のタスクである。主メモリ6からひとたびフェッチされたデータ項目は、補充バス32を介してキャッシュアクセス回路20に供給され、主メモリ内のアドレスと共にキャッシュメモリ22にロードされる。CPUが実行を継続できるように、データ項目自体もデータバス8に沿ってCPUへ戻される。上で簡単に述べた直接マップ・キャッシュメモリでは、主メモリ6から再び呼び出されたデータ項目およびそれのアドレスが、データ項目が検査のために最初にアクセスされた記憶場所にロードされることが明らかであろう。すなわち、そのデータ項目は、それを受けることができて、主メモリ内のラインインページ・アドレス中の最下位ビットセットに一致する行アドレスを持つ記憶場所にのみ重ね書きされる。もちろん、キャッシュメモリにもともと記憶されているデータ項目のページ数と、それにいまロードすべきデータ項目とは異なる。この「1対1マッピング」はキャッシュの有用性を制限する。

【0031】キャッシュシステムに一層高い柔軟性を持たせるために、nウエイセット連想キャッシュメモリが開発されている。4ウエイセット連想キャッシュメモリの例を図2に示す。このキャッシュメモリは4つのバンクB1、B2、B3、B4に分割される。それらのバンクは、図2に1つの行について概略的に示しているように、共通行アドレスにより行ごとに共通にアドレスすることができる。しかし、その行は、各バンクに1つずつ、合計4つのキャッシュエントリを含む。バンクB1のキャッシュエントリはバス26aに出力され、バンクB2のキャッシュエントリはバス26bに出力され、以下バンクB3、B4についても同様である。そうすると、1つの行アドレス（またはラインインページ・アドレス）に対して4つのキャッシュエントリが認められる。行がアドレスされるたびに、4つのキャッシュエントリが出力され、それらのエントリのアドレスの実ページ数が、翻訳ルックアサイド・バッファ10からの実ページ数と比較されて、どのエントリが正しいエントリであるかを判定する。このキャッシュに対してアクセスを試みたときにキャッシュミスがあったとすると、補充装置30が求められた項目を主メモリ6から検索し、たとえば、特定の項目がキャッシュに保持されている長さを基にする補充アルゴリズム、またはシステムの他のプログラム・パラメータに従って、その項目を1つのバンクの正しい行にロードする。そのような交換アルゴリズムは知られているのでここではこれ以上は説明しない。

【0032】しかし、nウエイセット連想キャッシュ（ここにnはバンクの数で、図2では4に等しい）は、シングル直接マップシステムの改良ではあるが、いぜんとして柔軟性に欠け、更に重要なことに、キャッシュの動作を正しく予測することができない。

【0033】ここで説明しているシステムは、より柔軟なキャッシュ補充システムによりコンピュータがキャッシュメモリを最適に使用できるようにするキャッシュ区画機構を提供するものである。

【0034】ここで説明しているシステムにおける翻訳ルックアサイド・バッファ10では、各TLBエントリに仮想ページ数と、実ページ数と、情報シーケンスとが関連づけられる。エントリの例を図3に示す。ここに、VPは仮想ページ数を表し、RPは実ページ数を表し、INFOは情報シーケンスを表す。情報シーケンスはメモリ内のアドレスについての種々の情報を知られているやり方で含む。そのやり方についてはここでは説明しない。しかし、ここで説明しているシステムによれば、情報シーケンスは区画標識PIを更に含む。その標識はここで説明している実施の形態では4ビット長である。したがって、情報シーケンスINFOのビット0ないし3は特定の標識を構成する。区画標識は、データ項目がキ

ャッシュメモリ２２に最初にロードされる時にそのデータ項目を置くことができる区画についての情報を与える。図２に示すキャッシュ構造では、各区画はキャッシュの１つのバンクを構成することができる。特定の標識では、各ビットはバンクの１つを指す。区画標識のビットｊにおける１の値は、そのページにおけるデータを区画ｊに置けないことを意味する。ビットｊにおける０の値は、そのページにおけるデータを区画ｊに置けることを意味する。区画標識の２つ以上のビットを０とすることにより、２つ以上の区画にデータを置くことができる。全部が０の区画標識では、データをキャッシュの任意の区画に置くことができる。全部が１の区画標識では、どのデータ項目もキャッシュメモリにロードすることができない。これは、たとえば、診断目的のために、キャッシュの内容をたとえば「凍結する」ために使用することができる。

【００３５】図３に示す例では、区画標識は、主メモリ内にその実ページ数を持つデータ項目の交換にバンクＢ１またはＢ３を使用できず、バンクＢ２またはＢ４を使用できることを示す。

【００３６】２つ以上のバンクをページに割り当てることは全く可能である。その場合には、ラインインページ・アドレスがそのキャッシュのための行アドレスより多くのビットを有するものとすると、区画はｋウエイセット連想キャッシュとして振る舞う。この場合にはｋ個の区画がページに割り当てられる。そうすると、ここで説明している例では、図３の実ページ数はバンクＢ２とＢ４を使用することができる。しかし、それはバンクＢ１とＢ３を使用することができない。

【００３７】区画情報はキャッシュルックアップには使用されず、キャッシュの交換または補充の場合のみに使用される。したがって、キャッシュアクセスがキャッシュメモリのどの場所に保持されているデータ項目も探すことができ、交換ではそのページアドレスに対して許されている区画内に交換するだけである。

【００３８】図４は補充装置３０の内容を一層詳しく示すものである。図４には補充バス３２を、データバス３２ａと、アドレスバス３２ｂと、交換情報を伝えるバス３２ｃとの３つの別々のバスとして示す。データバス３２ａとアドレスバス３２ｂはメモリアクセス回路５０に接続される。メモリアクセス回路５０はメモリバスを介して主メモリをアクセスする。交換情報は判定回路５２に供給される。判定回路は実アドレス３４と、バス３６における区画標識ＰＩと、ミス信号３８とをも受ける。判定回路５２は、主メモリからアクセスされたデータを置くべきキャッシュ内の正しい区画を決定する。

【００３９】区画標識ＰＩは他の任意のＴＬＢエントリのようにＴＬＢでセットすることができる。ここで説明している例では、ＣＰＵ２で実行しているカーネルモード・ソフトウエアによって区画標識はセットされ、特定のキャッシュ区画に置くべきでないページが、その区画のためにセットされたその特定のキャッシュ区画の区画標識ビットを持たないようにすることが、カーネルモード・ソフトウエアの責任である。しかし、ユーザーは、キャッシュ区画を変更することを求めることにより、区画を変更することができる。その場合にはＣＰＵ２はカーネルモードへ変更して要求を実現し、それに従ってＴＬＢエントリを変更し、その後でユーザーモードへ戻ってユーザーが継続できるようにする。このようにしてユーザーはキャッシュの区画の動作を変更して、従来可能であったものより高い柔軟性を持たせることができる。

【００４０】ここで説明しているキャッシュ区画機構はマルチタスクＣＰＵにとってはとくに有用である。マルチタスクプロセッサは２つ以上のプロセスを「同時に」実行することができる。実際には、プロセッサはプロセスの一部を実行し、何らかの理由、おそらくデータを必要とするとか、続行するための刺激を必要とするとか、でそのプロセスが中断された場合に、プロセッサは他のプロセスを直ちに開始するものである。したがって、このプロセスを停止して続行するためのデータまたは他の刺激を待つことができる場合でも、プロセッサは常に動作する。図５はそのような状況を線図的に示す。図５の左側に、種々のプロセスＰ１、Ｐ２、Ｐ３、Ｐ４を実行することをプロセッサが企てることができるシーケンスを示す。図５の右側に、それらのプロセスのデータがメモリに保持されていることをプロセスが予測できる場所を示す。そうすると、プロセスＰ１のためのデータがページ０に保持される。プロセスＰ２のためのデータがページ１と２に保持される。プロセスＰ３とＰ４のためのデータがページ３を共用する。この例では、プロセッサはプロセスＰ１の第１のシーケンスと、プロセスＰ２の第１のシーケンスと、プロセスＰ１の第２のシーケンスと、プロセスＰ２の第２のシーケンスとを実行し、その後でプロセスＰ３の第１のシーケンスを実行する。プロセッサはプロセスＰ１の第２のシーケンスが実行されると、プロセスＰ１が完全に実行されたことになる。従来のキャッシュシステムでは、プロセッサがプロセスＰ２の第１のシーケンスの実行を開始して、ページ２からのアクセスを求めると、それらのラインにおけるデータ項目と命令はキャッシュ内で、ページ０からの以前に記憶されたデータ項目および命令と交換することが容易にわかるであろう。しかし、プロセスＰ１の第２のシーケンスが実行されると、それらをまもなく再び求めることができる。

【００４１】ここで説明しているキャッシュ区画機構は、タイミングの遅れと、それに起因する不確実性とを避ける。図６はプロセッサがプロセスＰ１を実行中のキャッシュの区画と、プロセッサがプロセスＰ３を実行するために切り替えた時の区画の変更、等を示す。図６は各キャッシュのためのＴＬＢキャッシュ区画標識も示

す。したがって、図6の左側はプロセッサがプロセスＰ１とＰ２を実行している間に区画されたキャッシュを示す。プロセスＰ１はキャッシュのバンクＢ１とＢ２を使用できるが、バンクＢ３とＢ４は使用できない。逆に、プロセスＰ２はキャッシュのバンクＢ３とＢ４を使用できるが、バンクＢ１とＢ２は使用できない。これを下のＴＬＢエントリに見ることができる。すなわち、ページ０が、バンクＢ３とＢ４ではなくて、バンクＢ１とＢ２をアクセスできるようにするキャッシュ区画標識を有する。ページ１と２が、バンクＢ１とＢ２ではなくて、バンクＢ３とＢ４をアクセスできるようにするキャッシュ区画標識を有する。ページ３がキャッシュをアクセスすることを阻止するキャッシュ区画標識を有する。したがって、データ項目をプロセスＰ３からキャッシュにロードするというプロセッサによるどのような試みも禁止される。しかし、ここで説明しているプロセス・シーケンスでは、これは欠点ではない。その理由は、わかるであろうが、プロセッサはプロセスＰ１の実行を終了するまではプロセスＰ３のどの部分も実行しようとはしないからである。プロセスＰ３を実行しなければならないという何らかの理由で、プロセッサがそれを行ったとすると、ダウンサイドだけになって、直接メモリからそれのアクセスを行わなければならず、かつキャッシュの使用を許されない。

【００４２】プロセスＰ１の実行が終了すると、プロセッサがＴＬＢ内のキャッシュ区画標識を変更できるようにするために、プロセッサはカーネルモードを要求することができる。ここで説明している実施の形態では、カーネルプロセスはキャッシュをアクセスしない。その代わりに、カーネルプロセスは、区画標識がキャッシュの挙動を変更するためにＴＬＢエントリを変更する。この変更を図6の右側に示す。したがって、今はキャッシュ区画標識はプロセスＰ１がキャッシュを使用することを全く阻止するが、キャッシュのバンクＢ１とＢ２をアクセスできるようにキャッシュ区画標識を変更することにより、それらのバンクをプロセスＰ３とＰ４に割り当てる。そうすると、プロセスＰ３を実行することをプロセッサが予測すると、そのプロセッサは今はキャッシュ機能を有する。

【００４３】上記実施の形態のいくつかの可能な変更を以下に説明する。

【００４４】上記実施の形態では、ＣＰＵによりアドレスバス４に出されたアドレスが仮想ページ数４ｂと、ラインインページ４ａとに分割される。しかし、全体の仮想アドレスはＣＰＵからキャッシュのためのルックアップ回路に送られる。逆に、ＣＰＵは実アドレスをルックアップ回路へ直接送ることができる。重要なことは、キャッシュ区画標識が主メモリ内のアドレスと共同して提供されることである。上記実施の形態では、ルックアップおよび補充におけるキャッシュをアクセスするために、１つのキャッシュアクセス回路２０を示している。しかし、補充のために追加のアクセスポートをキャッシュに設けることも可能であるから、ルックアップと補充はキャッシュメモリ２２のための種々のアクセスポートを介して起きる。

【００４５】上記実施の形態では、補充装置３０とキャッシュアクセス回路２０を個々のブロックで示している。しかし、補充装置の機能とキャッシュアクセス回路の機能とを、ルックアップと補充を行う単一のキャッシュアクセス回路に組み合わせることが全く可能である。

【００４６】図2および図8からわかるように、キャッシュ内の各記憶場所は主メモリ内のアドレスと項目（データまたは命令）を保持する。メモリアドレスの全てをキャッシュ記憶場所に保持することが必要ではない。たとえば、アドレスの最上位ビットが全体として保持されて、そのキャッシュエントリのためのタイミングを構成する。これはこの技術で知られているためにここでは説明しない。

【００４７】次にキャッシュ・コヒーレンシー機構について説明する。データおよびメモリキャッシュの状態を変更するオペレーションのタイプを表１に示す。

【００４８】

【表１】

オペレーションタイプ	効　果	動　作
フラッシュ	汚れたデータが他のユーザが見えるようにする	キャッシュ内の最後に書込まれた値をメモリに書込む
除去	キャッシュからデータを除去する	キャッシュをクリヤする
ＩＣｏｈｅｒｅ	古いデータを命令キャッシュから除去する	キャッシュをクリヤする

データコヒーレンシー・オペレーションは、オペレーションの範囲内で主メモリ内のアドレスでメモリ内の項目をアクセスするロードおよびストアに関して命令される。たとえば、アドレス空間の重なり合う部分への「フラッシュ」ご続く「ストア」により、新たに書込まれたデータについてフラッシュが確実に動作するようにする。動作はユーザーモードで実行される。ユーザーモード・スレッドが、オペレーションを実行すべきであるページについての読出し許可または書込み許可をとらなければならない。

【００４９】ラインおよび区画について動作するために各コヒーレンシー・オペレーションが提供される。ラインオペレーションはここでは完全にするために説明するが、本発明の構成部分ではない。

【００５０】＜ラインオペレーション＞ラインオペレーションにより個々のラインのキャッシュ制御を行えるようにされる。ラインはラインに含まれている任意のバイトアドレスにより指定される。それらの命令は、所与のアドレス範囲において動作する命令の最適化できるループの構成を容易にすることを意図している。たとえば、キャッシュから所与のバッファを除去するために、除去ライン命令がループにおいて実行され、第１のアドレスオペランドがバッファのスタートであり、次のアドレスが次のキャッシュラインのスタートアドレスである。アドレスがバッファの終りを超過するとループは終わる。キャッシュラインのサイズはインプリメンテーションにより決定される。ここで説明している実施の形態ではそのサイズは３２バイトである。

【００５１】＜区画オペレーション＞区画を基にした命令は、１組の区画のうちのどの１つに対して働きかけるかを、関連するＴＬＢエントリを介して、決定するためにアドレスを用いる。その後で命令は、アドレスを交換できるそれらの区画のキャッシュの内部のラインについて働きかける。

【００５２】区画を基にした命令は一連の命令において使用することを意図するものである。下記の諸条件は、アドレスを含んでいるページを交換できる全区画に対してオペレーションを実行する。

・　アドレスはページ内の最初のバイトのアドレスに対して初期化される。

・　データを常駐できる各区画について、区画オペレーションは同じアドレスで１回反復される。

・　ページ内のオフセットが区画サイズに達するまでアドレスは増加される。区画オペレーションを各増加ごとに反復する。

【００５３】ここで説明している実施の形態では、最小のページサイズは区画サイズと少なくとも同じである。したがって、単一のページについてのオペレーションを用いることは全区画について働きかけることである。

【００５４】ソフトウエアがページ内の区画サイズより大きいバイトのアドレスを用いるものとすると、これは、既に働きかけられた、キャッシュ内のラインにマップする。したがって、コードは正しく実行するが、そうすると不必要な命令の実行において性能が低下するという不利益をこうむる。

【００５５】区画を基にした命令は、キャッシュ内の１組のラインを識別するためにアドレス（および、したがって、区画識別子）を用いる。その後で、指定されたアドレスがキャッシュに保持されている主メモリ内のアドレスに一致しようがしまいが、その１組のラインの１つに対して働きかけられる。ＰＩビットがクリヤされている区画のみがそれらの命令により変更される。すなわち、実行中のプロセスがアクセスする区画のみに対してコヒーレンシー命令が働きかける。

【００５６】このキャッシュ・コヒーレンシー機構は下記の命令を提供する。それらの命令では、ｄｍｅｍという記号がコンピュータシステムの主メモリを指す。

【００５７】＜Ｆｌｕｓｈ＞それらの命令は、汚れたデータを他のユーザーが確実に見ることができるようにするために提供される。すなわち、関連する記憶場所に保持されている項目が、項目を有するそのキャッシュ記憶場所に保持されている主メモリ内のアドレスにライトバックされる。

【００５８】

【表２】

ｆｌｕｓｈｌｉｎｅ

ラインをフラッシュする ベース、オフセット	符号なし（ｘ）　符号なし（ｘ）
ｄｍｅｍ［ｂａｓｅ＋ｏｆｆｓｅｔ］を含んでいるラインへの以前の全ての書込みが、このデータを共用する他のユーザーが見ることができるようにする	

【００５９】

【表３】

flushpart

ラインをフラッシュする ベース、オフセット　　符号なし（x）　符号なし（x）
dmem［base+offset］に対するメモリアクセスにより交換できる汚れたキャッシュラインをフラッシュする。 ラインをフラッシュすることによって、このデータを共用する他のユーザーが見ることができるようにする。 交換できる全てのラインがクリーンであれば命令は効果がない。

＜Purge［データ項目に対して］＞キャッシュからデータを除去するためにそれらの命令が提供される－それらの命令はキャッシュ内のデータ項目を、それらの項目で指定された主メモリ内のアドレスにライトバックする。

【0060】

【表4】

purgeline

ラインを除去する ベース、オフセット　　符号なし（x）　符号なし（x）
dmem［base+offset］を含んでいるライン内の任意の汚れている項目をメモリにライトバックし、あらゆる場合にラインを無効にする

【0061】

【表5】

purgepart

区画を除去する ベース、オフセット　　符号なし（x）　符号なし（x）
dmem［base+offset］に対するメモリアクセスにより交換できる有効なキャッシュラインを除去する。 ラインが含んでいる任意の汚れたデータをメモリに書込むことによりそのラインは除去される。 交換できる全てのラインが無効であれば命令は効果がない。

＜ICohere［命令に対する］＞後の命令フェッチが古いデータを命令キャッシュから読出す内容にするためにそれらの命令が提供される。

【0062】

【表6】

icohereline

命令キャッシュ内のラインをコヒーレントにする ベース、オフセット　　符号なし（x）　符号なし（x）
imem［base+offset］を含んでいる命令キャッシュラインを無効にする。

【0063】

【表7】

icoherepart

命令キャッシュ内のラインをコヒーレントにする ベース、オフセット　　符号なし（x）　符号なし（x）
imem［base+offset］に対するメモリアクセスにより交換できる有効な命令キャッシュラインを無効にする。 imem［base+offset］をキャッシュできないならば、命令は効果がない。 交換できる全てのラインが無効であれば命令は効果がない。

他の実施の形態では、区画をベースとするフラッシュ命令は下記のフォームを持つことである。下記の命令では、var<a：b>は可変varのビットaないしb

flushpart

区画をフラッシュする ベース、オフセット　　符号なし（x）　符号なし（x）
addr←base+offset addrφ<12：63>←addr<12：63> index＝φであれば、index<4096，index+＝32 addrφ<0：11>＝index<0：> dmem［addrφ］に対するメモリアクセスにより交換できる複数の汚れたキャッシュラインをフラッシュする。

単一の命令がキャッシュ内の複数のライン働きかける場合に、除去命令およびコヒーレンスでない（incohere）命令が類似の形態をとることができる。

【0065】明らかに、以上の説明は、キャッシュが区画されない場合、すなわち、全体のキャッシュが単一の区画であるとみなされる場合、にあてはまる。

【図面の簡単な説明】

【図1】キャッシュシステムを組み込んだコンピュータのブロック図である。

【図2】4ウエイセット連想キャッシュを示す略図である。

【図3】翻訳ルックアサイド・バッファにおけるエントリの例である。

【図4】補充装置のブロック図である。

【図5】マルチタスクプロセッサの動作を示す線図である。

【図6】図5に示すシステムのキャッシュ動作の変更を示す線図である。

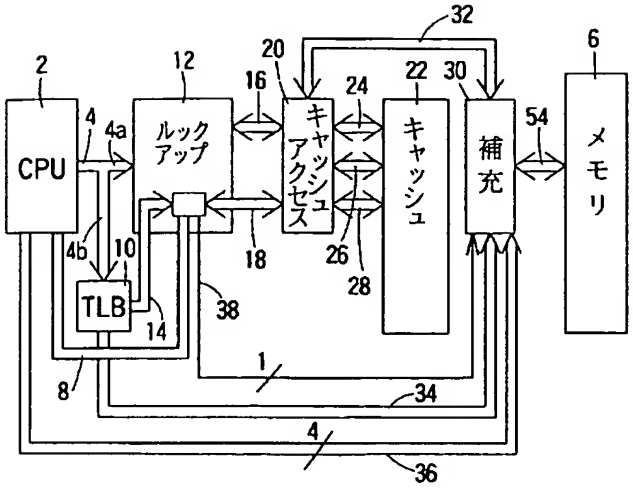
【図7】自動コヒーレンシー制御のブロック図である。

【図8】「古い」データ項目と「汚れた」データ項目を示す。

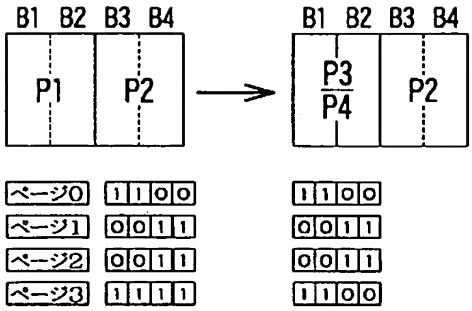
【符号の説明】

2　CPU
4，24　アドレスバス
6　主メモリ
8　データバス
10　翻訳ルックアサイド・バッファ
12　ルックアップ回路
20　キャッシュアクセス回路
22　キャッシュメモリ
26　データバス
28　制御バス
30　補充装置
32　補充バス
36　4ビットバス

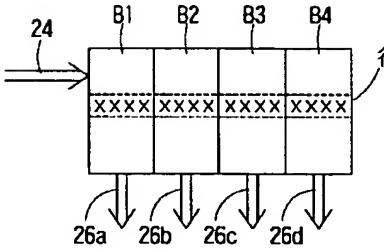
【図1】



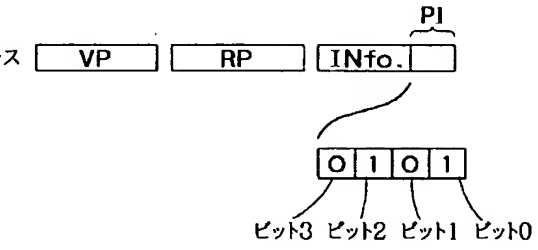
【図6】



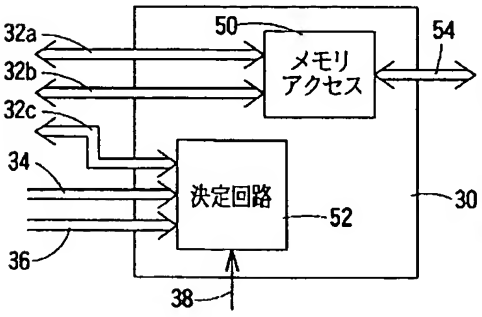
【図2】



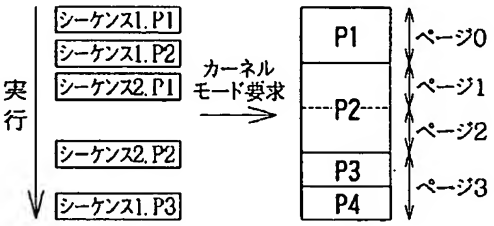
【図3】



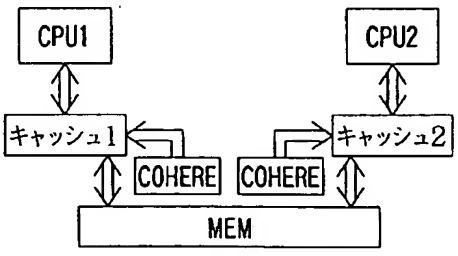
【図4】



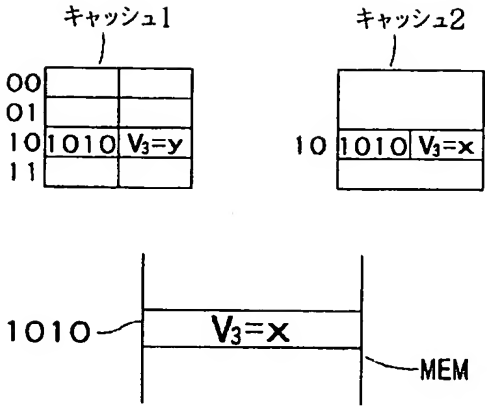
【図5】



【図7】



【図8】



フロントページの続き

(72)発明者　グレン、ファーラル
イギリス国ブリストル、ロング、アシュトン、ロング、アシュトン、ロード、157

(72)発明者　ブリュノ、フェル
フランス国サスナージュ、リュ、デュ、ムーシュロット、14